

PATENT ABSTRACTS OF JAPAN

(11)Publication number : **64-026969**

(43)Date of publication of application : **30.01.1989**

(51)Int.Cl.

G06F 15/20

(21)Application number : **63-155114**

(71)Applicant : **AMERICAN TELEPH & TELEGR
CO <ATT>**

(22)Date of filing : **24.06.1988**

(72)Inventor : **AGRAWAL PRATHIMA
DALLY WILLIAM J
KRISHNAKUMAR ANJUR S**

(30)Priority

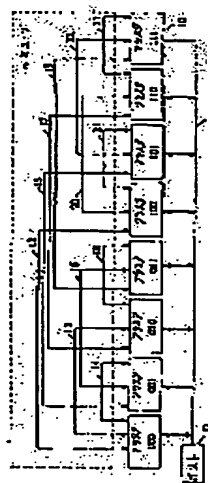
Priority number : **87 66921** Priority date : **25.06.1987** Priority country : **US**

(54) PROGRAMMABLE ACCELERATOR AND ITS METHOD

(57)Abstract:

PURPOSE: To enable programming and also to enable a high-speed asynchronous communication to be executed by utilizing the plural same processing elements which are flexibly and mutually connected so as to form a cluster through a switch in a simulation hard accelerator.

CONSTITUTION: The accelerator consists of the eight clusters 10 and a host computer 11 which are mutually connected to a Boolean 3 cube with two-way lines 12-24. They are mutually connected with a common two-way bus 11. Each cluster has a binary address corresponding to the coordinate of the cluster in the n-cube (a solid) in n-space. A task is down-loaded from the host to the cluster with the bus 25. The bus 25 provides direct access to a memory inside the cluster and the buses 12-24 execute communication by a message. The message adds the cluster address so that the n-cube is connected only to the adjacent cluster.



⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭64-26969

⑬ Int. Cl.⁴

識別記号

庁内整理番号

⑭ 公開 昭和64年(1989)1月30日

G 06 F 15/20

D-7230-5B

審査請求 未請求 請求項の数 7 (全15頁)

⑮ 発明の名称 プログラマブル アクセラレータ及びその方法

⑯ 特 願 昭63-155114

⑰ 出 願 昭63(1988)6月24日

優先権主張 ⑱ 1987年6月25日 ⑲ 米国(US) ⑳ 066,921

㉑ 発 明 者 ブラシマ アグラワル アメリカ合衆国 07974 ニュージャージー, ニュープロ
ヴィデンス, コルチエスカー ロード 40

㉒ 発 明 者 ウィリアム ジェー, アメリカ合衆国 01701 マサチューセッツ, フラミンガ
ム, トーリー ホー レーン 11

㉓ 発 明 者 アンジュア エス. ク アメリカ合衆国 07060 ニュージャージー, ウオーレ
リシユナクマー ン, マウント ホレブ ロード 205

㉔ 出 願 人 アメリカン テレフォ アメリカ合衆国, 10022 ニューヨーク, ニューヨーク,
ン アンド テレグラ マデイソン アヴェニュー 550
フ カムパニー

㉕ 代 理 人 弁理士 岡部 正夫 外3名

明 細 書

1. 発明の名称

プログラマブル アクセラレータ及びその
方法

2. 特許請求の範囲

1. プログラマブル アクセラレータにおい
て、該アクセラレータが：

情報を該アクセラレータに向けて或はこれ
から伝送するための通信網；及び該通信網に
応答する複数のプログラマブル処理要素を含
み、個々の該処理要素が

複数のデータ バス

該バスと該通信網の間でメッセージを通信
するための入力及び出力キュー ブロック、
及び

該バス上の信号にตอบสนองして1つのバスから
選択された1つの欄及びもう1つのバスから
選択されたもう1つの欄を選択し、これら選
択された欄に関して操作し、この結果を該バ
スの1つの上の上のあらかじめ選択された欄内に

置くための欄操作ユニットをもつことを特徴
とするプログラマブル アクセラレータ。

2. 該キューが空或はフルのとき選択された動
作を中断するためのストール手段が更に含ま
れることを特徴とする請求項1記載のプログ
ラマブル アクセラレータ。

3. 1つのメモリ及び該メモリを任意の所望の
語幅に構成するための手段が更に含まれるこ
とを特徴とする請求項1記載のプログラマブ
ル アクセラレータ。

4. ハードウェア アクセラレータにおいて、
該アクセラレータが：

第1の複数のクラスタ；

第2の複数のノード及び第3の複数の通信
リンクをもち、該リンクが該ノードを相互接
続し、該ノードが該クラスタを該リンクに接
続する通信網；

該ノードに接続されたクラスタに対する入
り信号の所定の経路を該入り信号の所定の第
一優先に基づいて評価するためのコントローラ

要素、

該コントローラに加えられた該信号の経路を該信号の該着信先に基づいて決定するためのスイッチ、及び

該スイッチにตอบสนองして該信号を該ノードに接続された出リンク及び該ノードに接続された該クラスタに送るためのバッファを含み；

該クラスタが更に

専用のランダム アクセス メモリをもつ複数のプログラマブル処理要素、及び

該プロセッサを1つのノード及び互いに相互接続しパイプライン構成を形成するためのスイッチを含むことを特徴とするプログラマブル アクセラレータ。

5. グラフ処理用のプログラマブル アクセラレータにおいて、該アクセラレータが：

複数の実質的に同一のプログラマブル処理要素；

該処理要素をパイプライン構成に相互接続

ンを展開するためにシミュレーション アルゴリズムを実行するための方法において、該個々の処理要素がデータの選択された欄に関して動作するように設計され、該欄が語の一部であり；該方法が

該処理要素をパイプラインに構成するステップ、及び

個々の該処理要素にて該シミュレーション アルゴリズムのあらかじめ指定された部分を実行するステップを含むことを特徴とする方法。

3. 発明の詳細な説明

発明の技術分野

本発明は一般的にはコンピュータ シミュレーションに、より具体的には回路設計のコンピュータ シミュレーションに用いられる通常ハードウェア アクセラレータと呼ばれる専用高速プロセッサに関する。

発明の技術的背景

ハードウェア アクセラレータは特定のアル

ゴリズム、該処理要素の間でメッセージをバスするための通信網；及び

該処理要素の個々にプログラム及びアクセラレータによって実行されるべき所定のタスクの特性であるデータ情報をロードするための手段を含むことを特徴とするプログラマブル アクセラレータ。

6. グラフ処理用のプログラマブル アクセラレータにおいて、該アクセラレータが：

複数の処理要素；及び

語から成るメッセージを介して該処理要素との及び処理要素間の通信を行なうための手段を含み、該個々の処理要素がデータの選択された欄に関して動作するように設計され、欄が語の一部であることを特徴とするプログラマブル アクセラレータ。

7. 複数の実質的に同一の処理要素、及び語から成るメッセージを介して該処理要素との及び該処理要素間の通信を行なうための手段を含むシステムにて論理回路のシミュレーション

ゴリズム或はあるクラスのアルゴリズムの効率的な使用のために特別に設計された専用処理装置である。これら要素は広範囲のコンピュータ支援設計問題に対する実現可能な解決としてますます注目を集めており、これらの使用は結果として顕著な性能の向上に結びついている。今日のシミュレーション要素を用いた場合は、秒当たり10⁴以上のゲート評価を遂行することが可能であり、これはソフトウェア論理シミュレーションと比較して桁違いの向上である。

シミュレーション要素はこれらの高レベルの性能を1つ或は複数の以下の手段を遂げて達成する。つまり、複数の同一ハードウェア ユニットが用いられ、これらの間でシミュレーション問題が分割され、動作がこれらユニットを通じてパイプライン化され、また手元のアプリケーションに合せられた専用論理、及び処理ユニット間の専用の相互接続が用いられる。ハードウェア アクセラレータに関しての貴重な調査がトム ブランク(Tom Blank)によってIEE

EE コンピュータの設計及びテスト (IEEE Design and Test of Computers) . 1984年8月号に掲載の論文【コンピュータ支援設計に用いられるハードウェア アクセラレータの調査 (A Survey of Hardware accelerators used in Computer-Aided Design)】に紹介されている。更に適当な資料として、【論理シミュレーション マシン (A Logic Simulation Machine)】、マイロン アブラモビシ (Miron Abramovici) . IEEE . Vol - CAD - 2 . No . 2 . 1983年4月号；【データ フロー アーキテクチャーを用いての論理シミュレーションのハードウェア アクセラレーション (Hardware Acceleration of Logic Simulation using a Data Flow Architecture)】、IEEE . 1985；【VLSI 設計におけるシリコン溶融彫刻 ナイチェ (Silicon Solutions Carves Niche in VLSI Design)】、エレクトロニクス . 1985年8月12日号；【パワースパイスによる回路のより高速より正確なシミュレーション

(Powerapic Simulates Circuits Faster and More Accurately)】、エレクトロニクス . 1985年8月28日号；及びスイッチレベル シミュレーションに対するハードウェア アーキテクチャー (A Hardware Architecture for Switch-Level Simulation) . W . J . ダリー (W.J.Dally) その他、IEEE . 1985年を挙げる事ができる。

これら先行技術によるシステムに用いられるハードウェア アクセラレータの殆どは、特定のタスクを非常に効率的に遂行するために専用化された多くの処理要素、或は要素を含み、これら要素が密着した相互作用システムを形成するように相互接続される。周知のシミュレーション要素の全てが共通にもつ短所は、これらが処理要素間の固定された相互接続を採用し、これによって、柔軟性が失われ、また処理要素自体が特定のアプリケーションに専用化され、このため異なるアプリケーションを扱う場合に異なる設計が必要となることである。

本発明の概要

本発明はスイッチを通じてクラスタを形成するようにフレキシブルに相互接続された複数の実質的に同一の高速専用処理要素 (processing element, PE) を採用することによって先行技術による回路シミュレーション技術と比較して大きな向上を達成する。この処理要素 (PE) がプログラマブルであるのに加えて、この相互接続スイッチは処理要素間の信号の経路の動的切り換えを可能とする。この処理要素は要素間の高速非同期通信を可能とするキュー ユニットを含む。本発明においては、複数のクラスタが n -立体構成に相互接続され、クラスタの全てが1つのホスト コンピュータと通信するが、この階層構成によって追加の利点が得られる。

実施例

第1図は本発明による高速シミュレーション用マイクロプログラマブル アクセラレータ (Microprogrammable Accelerator for Rapid

Simulation, MARS) の一般ブロック図である。これは双方向ライン12-24を介してブーリアン 3-キューブ (Boolean 3-cube) に相互接続された8個のクラスタ10と1つのホスト コンピュータ11を含む。これらはまた共通の双方向バス11を介して相互接続される。例々のクラスタは (n -空間内の) n -キューブ (立体) 上のそのクラスタの座標に対応する2進アドレスをもつ。第1図は3-キューブ相互接続を示すが、更に高次元にするため (15個の処理要素をもつ) ブーリアン 4-キューブに拡張することも簡単である。第2図は相互接続スイッチのノードを示す。ブーリアン n -キューブの説明に関しては、例えば、W . D . ヒリス (W.D.Hillis) . サイエンティフィック アメリカーン (Scientific American) . 1987年6月号 (ページ108-115) による "接続マシン (The Connection Machine)" にみられる。

第1図のシステム内の例々のクラスタは n -

キューブへのインタフェース ノード、メモリと関連する複数の処理要素、及び相互接続スイッチを含む。

動作において、MARSによって遂行されるべきシミュレーション タスクはホストプロセッサ11からクラスタにバス25を介してダウンロードされる。バス25はクラスタ内のメモリへの直接アクセスを提供し、バス25を通じてデータを通信するためのプロトコールは通常のDMA (direct memory access、直接メモリアクセス) プロトコールである。一方、バス12-24はメッセージにて通信する。個々のメッセージは1つの見出し語及びこれに続く1から16の間のデータ語から成り、本実施態様における個々の語は16ビット長である。個々の語は事前に選択された情報の欄(Field)を異なるビット増分にて含む。見出し語はメッセージ長、クラスタ アドレス、メッセージがそれに向けられる着信先クラスタ内の特定の処理エンジン指定する。クラスタアドレスはキューブ

が隣接クラスタのみに接続を提供するために必要である。隣接クラスタ以外への接続が要求される場合は、複数のノード インタフェースを通じての“ディジー チェーン(daisy chain)”接続が必要である。

第2図は処理要素とキューブ相互接続網の間のクラスタ ノード インタフェースを示し、ここで、 n は3である。これは4つの入力及び4つの出力をもつクロスバースイッチ30、個々の入力に接続された入力コントローラ31、及び個々の出力に接続された出力バッファ32を含む。1つの入/出力ペアがノードの所の処理要素とインタフェースするために割り当てられ、一方他の3つの入/出力ペアは信号キューブ相互接続網に沿って送るために用いられる。入力コントローラは従来のアドレス計算回路であり、個々は自体のアドレス(そのコントローラが位置するノードのアドレス)を認識している。コントローラは入りメッセージの所望のクラスタ アドレスを読み出し、所

望のアドレスを自体のアドレスと比較することによって(ビット ワイズ排他的OR)メッセージがそれにスイッチされるべき好ましい出力リンクを計算し、結果としての経路情報をメッセージとともにクロスバースイッチ30に加える。バッファ32は着信先クラスタがビジーの場合、或は所望の経路上のリンクがブロックされた場合、メッセージが待ち行列上に置かれることを保証する機能をもつ。

第3図は第1図のクラスタ10のアーキテクチャを示す。第3図の構成は14個の処理要素(PE-0からPE-13)を含み、個々の処理エンジンは関連するランダム アクセスメモリ(RAM-0からRAM-13)を含む。個々のPEはそれと関連するRAMと専用双方向バス(例えば、33)を通じて通信し、これに加え、個々のPEはスイッチ40と通信する。RAMはまた双方向バス41と通信する。バス41はインタフェース回路42を通じて、ホスト プロセッサ11とバス25を介し

て通信する。こうして、個々のPE/RAMセットはバス41及びスイッチ40の両方に接続される。バス41にはまたハウスキーパー ブロック43、クラスタRAM44、及び大容量記憶のためのローカル ディスク メモリ45が接続される。ブロック43はスイッチ40にも接続される。最後に、スイッチ40もコントローラ31に接続され、コントローラ31はクラスタをキューブ網に接続する。

スイッチ40はPEをラウンドスปีカ ブロック及び網インタフェースと接続するローカル メッセージ スイッチである。これは、第4図に示されるように、クロスポイントスイッチ(41)のクロスバース構成である。より具体的には、第4図は(ソース)PEからのロウに沿っての16個のデータ ラインの1つ(ライン51)、別の(着信先)PEからのカラムに沿っての16個のデータ ラインの1つ(ライン52)、及びソースを着信先に接続する1つのクロス ポイント41を示す。このデータ

ラインに加えて、ソースPEは、4つのアドレスライン、1つの出力要求(REQ)ライン、1つのアクノレッジライン(84)、及び1つの要求/アクノレッジ出力ライン(RA-アウト)を与える。着信先PEはRA-アウトラインに対応する要求/アクノレッジ入力ライン(RA-アウト)を含む。クロクポイント41の個々のコラムはまた1つの優先ライン及び1つのロック指標ラインを含む。この優先ラインは、衝突が起った場合、どのソースPEが他のソースPEに優先して所望の着信先PEへのアクセスを得るべきかの決定ができるように、PEに対する優先割り当てを設定する。ロック指標ラインはソースPEが着信先PEへのアクセスをいったん得ると、このアクセスがソースPEが接続を放棄することを選択するまで維持されることを保証する。

動作において、アドレスラインがロウ内の個々のクロスポイント41内でプリワイヤード複号器53によって複号される。適当なクロス

も接続を確立していないことを意味する)、REQパルスはゲート55と無関係にラッチ58から56に届き、GRANT信号が高値にラッチされる。優先ラインはクロスポイント41からスイッチ61を通じて出る。スイッチ61は優先ライン出力をGRANT信号が高値であるか低値であるかによって、優先ライン入力かアースのいずれかに接続する。優先ラインをアースすることは、そのコラム上のその後のクロスポイントがGRANT信号を与えられることを防止する。同様に、ロック指標ラインはスイッチ60によって分岐される。これはコラム内の任意のクロスポイント41にGRANT信号が与えられたときアースレベルにされ、これはそのコラム内の他の全てのクロスポイントがGRANT信号を得ることを防ぐ。

GRANT信号がクロスポイント41に与えられると、ゲート54のREQ信号出力はこの情報をライン37上のアクノレジメント信号としてソースPEにバズする。また、ソースP

ポイントの所で、この複号されたアドレスがREQライン(ゲート54内)を起動し、信号がゲート55及び56に送られる。ゲート55及び56の出力はゲート57に加えられ、これはGRANT信号を生成するためにラッチされる。GRANT信号はゲート55に加えられ、従って、GRANT信号が高値のとき、REQパルスが出現し、このパルスがゲート55及び57を介してラッチ58に送られ、ここでこれはGRANT信号のその高値の状態に保持するために補強される。GRANT信号が高値でないときは、ゲート55は不能にされる。

ゲート56は、ゲート54によって制御されるのに加えて、ロック指標ライン及び優先ラインエンタリングクロスポイント(priority line entering crosspoint) 41によって制御される。エンタリング優先ラインが高値であるかぎり(これはより高い優先PEが接続を求めていることを意味する)そしてロック指標ラインが高値であるかぎり(これは他のどのPE

Eのデータラインがスイッチ35を通じて着信先PEのデータラインに接続され、RA-アウトラインがスイッチ38を通じてRA-インラインに接続される。

スイッチ60、61、35、38が第4図にGRANT信号が高値のときに対応するモードにて示される。

RA-イン及びRA-アウトラインが第4図にスイッチ38を介して相互接続される単一ラインとして示される。機能的には、これらは4つの双方向ラインとみなすことができる。但し、実際には、これらの機能は1つのライン、スイッチ38へのラッチング、及び1つの2フェーズクロックにて実現できる。RAラインはソースと着信先PEとの間の通信を許す機能をもつ。単一ラインとの動作においては、クロックの最初のフェーズにおいて、ソースPEはRA-アウトライン上に着信先にメッセージが送られてくることを知らせる1つの信号を送る。同時に、RA-インは着信先PEがそ

の入力キュー(input queue)内に空きをもつか否かを示す1つの信号を送くる。第2のクロックパルスにおいて、R A -アクト信号が着信先P BのR A -インラインに到達し、ここでこれは送られた情報を入力キュー内に書き込むのに用いられる(入力キュー内に空きがあると)。同時に、キュー内に空きがあり、データが受け入れられたという事実がソースP Bに到達し、ここでこれはメッセージ受信のアクノレージメントとして用いられる。

P Bは本ハードウェアアクセラレータの回路シミュレーションを遂行する。個々のP Bは複数の論理的に分離可能なユニット、例えば、事象スケジューラ、ファンアウトリストプロセッサ、機能評価ユニット等を含むシミュレーションパイプラインの1つの段として働くようにプログラムされる。ハウスキーパーブロック43は、従来のマイクロプロセッサ、例えば、モトローラM 68020 (Motorola 68020)である。これは基本的にシミュレーション

情報をP B (回路及びI/Oベクトル)内にロードすること、及び例外的な事態の扱い、例えば、システムクラッシュ後のタスクのチェックポイントからの再開の任務をもつ。ハウスキーパーはクロック内の個々のP Bのローカルメモリにバス41を介して直接にアクセスでき、またスイッチ40を介してP Bにメッセージを送ることができる。ハウスキーパーは回路区画(circuit partitions)を格納するためにディスク45を用いる。

本発明によるアクセラレータの最も重要な属性の1つは、クラスタ内の全てのP Bが同一のアーキテクチャー設計をもち、また個々のP Bがアプリケーション問題を解くためのパイプライン段として機能するように専用化されたマイクロプログラマブルプロセッサであることである。このアーキテクチャーは、第5図に示されるように、命令をレジスタ80を通じて復号器ユニット82-86に分配するマイクロプログラムRAM 71を含む。ユニット82-86

は、一方、ユニット72-76を制御する。つまり、外部(データ)RAMインタフェースユニット72、レジスタアレイユニット73、個体操作ユニット(field operation unit, FOU)74、アドレス演算ユニット(AAU)75、及びメッセージキューユニット78を含む。

プログラムが外部ソースからデータバス92(16ビット)及びアドレスバス93(16ビット)を介してRAM 71にロードされる。アドレスバスは外部アクセスライン(External Access line)94上に出現する信号の制御下においてマルチプレクサ87に接続する。ライン94はマルチプレクサ87に接続されているのに加えて、RAMインタフェースコントロール82に接続される。ブロック82と関連してこの機能はRAM 71がロードされているとき、RAMインタフェース72を外部から加えられた信号から分離することである。通常の動作においては、マイクロ命令R A

M 71は64ビットマイクロ命令をレジスタ80にくばる。この64ビットはレジスタ80によってさまざまなユニット、つまり、条件ストール及びストラップ論理ブロック81、RAMインタフェース制御ブロック82、レジスタ復号器83、FOU復号器84、AAU復号器85、及びキュー復号器86にくばられる複数の個をもつ。要素82-86は受信されたこれらの個を復号し、後に詳細に説明されるように要素72-76に夫々適当な制御信号を加える。ブロック81もさまざまなアラーム信号、例えば、キューユニット78からのQ E M P T Y及びQ F U L L信号に回答し、この機能はアラームが発生したときP Bの正常の動作を変更することにある。例えば、キュー78が一杯で、書き込み命令が発生した場合、ブロック81はこの書き込みを阻止するためクロックの最後のフェーズを不能にする。

ユニット72-76は3つのフェーズから成るマイクロ命令サイクルにおいて16ビット

バス77、78、及び79を通じて通信する。フェーズ1において、データがユニット73内のレジスタからバス上に読み出される。機能ユニットAAU(75)及びFOU(74)はフェーズ2においてデータに関して動作し、これらの結果をフェーズ3においてバス上に送る。バスの内容がフェーズ3において選択されたレジスタに書き込まれる。

レジスタ アレイ73は32個のレジスタ(R0-R31)を含む。殆どはそれらの内容をゲートし、クロックのフェーズ1において3つのバスの1つに加えることができる。これらバスの任意の1つの上の信号はクロックのフェーズ3の間にこれらレジスタ内に格納できる。幾つかのレジスタは追加の目的、例えば、メモリのアドレッシング、メモリ或はメッセージキューへのポートとしての働き、或は機能ユニットの構成などの目的を果たす。

レジスタ アレイ73内のさまざまな専用レジスタの説明に入る前に、PEのパラレルア

(branching situations)において、両方のアドレスがターゲット アドレスにロードされ、分岐決定に達したとき、次のRAM71アドレスを決定するのにクロック サイクルが消費されないようにされる。換言すれば、高速分岐が通信バスを用いることなしに達成される。バス91にR2が加えられるか或はR3が加えられるかは、復号器83によって制御される。バス91はアドレスをRAM71にマルチプレクサ87を通じて送る。マルチプレクサ87は、上に説明のごとく、ローディング プロセスの間はバス94を選択するが、通常の動作においてはバス91を選択する。

データRAMインタフェース ユニット72は2つのレジスタの連結によってアドレスされる。レジスタR0或はR1はライン99上のアドレスの16の最下位ビットを供給し、一方、レジスタ10或は11はライン96上のアドレスの8個の有効ビットを供給する。ライン99及び98によって供給される24ビット アド

レキテクチャーは(これは異なる経路を通じて互いに通信でき、また幾つかのユニットはパラレル経路を通じて通信できるさまざまなユニットをもつことを特徴とするが)ある1つの特定機能に対して1つ以上のレジスタを使用することを示唆し、結果として利用可能なパラレルizm(parallelism)がより良く活用できることを意味する。これ及び後に明白となるその他の理由から、後に説明される多くの機能はアレイ73内の2つのレジスタによって処理される。

レジスタ アレイ73内のレジスタR2及びR3は、例えば、プログラム アドレス レジスタであり、またこれらは経路91を介してRAM71にアドレスするのに用いられる。この2つのレジスタは2つの目的に供される。第1に、片方のレジスタがRAM71にデータを検索する指令を行なっている間、他方のレジスタに次のRAM71アドレスがロードできるように(或は単にプログラムのシーケンスを通じて増分できるように)される。第2に、分岐状況

レスは、適当なときに、夫々インタフェースユニット72の出力の所のライン89及び90に加えられる。データはバス88とバス77、78、及び79の間をコントロール ユニット82からの信号に回答して送られる。レジスタ アレイ73もバス97上の外部RAMインタフェース ユニット72に信号を供給する。メモリから端を発するバス97はアレイ73内のレジスタR6或はR7を選択し、このバス上の信号は外部メモリの有効的な“アスペクト比(aspect ratio)”を構成するためにRAMインタフェース ユニット(72)を制御する。つまり、ライン97はインタフェース ユニット72を制御し、ライン88-90に接続された任意の外部メモリ、例えば、個々が16ビットの84K語メモリがPEに対してあたかもこれが個々が18ビットの84K語、個々が1ビットの1M語、或はこの間の任意の他のアスペクト比を持つものとみえるようにされる。24ビット アドレス(ライン99及び96)を用い

て、そのPEと関連する外部メモリは読出し及び書込みの目的であたかもこれがレジスタアレイユニット73内のレジスタR29であるかのようにアクセスされる。メモリは個々のサイクルにおいて読出し或は書込みできるが、メモリアドレスを正しく設定することが必要である。

キューユニット75は実際には2つのユニット；つまり、1つの4語入力キュー、及び1つの4語出力キューである。両方とも需要PIFOメモリであり、両方ともレジスタ31としてアドレスされる。アレイユニット73内のレジスタR14はメッセージ着信先アドレスをライン95を通じて直接にキューユニット75に供給する。このアドレスはキューユニット75の出力の所のバス62にメッセージデータがバス98に配られるのと同時に配られる。RA-イン及びRA-アウト信号も又キューユニット75によってライン86及び87上に配られ、これらラインは一般にローカ

ルメッセージスイッチ40に送られる。QEMPTY及びQFULLライン(68及び69)はブロック81に接続されたアラーム信号であり、これは夫々出力キューがフルであるか否か、或は入力キューが空であるか否かを示す。こうして、R31への書込みによってメッセージが出力キューに送られ、一方、R31からの読出しによって入力キューからメッセージ語が受信される。

レジスタアレイユニット73内のレジスタR30は非存在(non-existent)であり、このアドレスは不用データ(unwanted data)に対する情報シンクとして機能する。この能力はこれによると命令を常にソースアドレスと着信先アドレスによって表わることができるために採用されている。

レジスタアレイ73内に提供されるもう1つの追加の専用レジスタは欄選択レジスタ(field select register, FSR)、つまり、レジスタR4或はR5である。このレジスタは

その入力及び出力ビット欄の幅及び開始ビット位置を選択することによってバス63を介して欄動作ユニット74を構成する。

FOUユニット74はビット欄動作(bit field operation)を遂行する。つまり、これはその2つのソースオペランドからビット欄を抽出し、これらビット欄に関して操作し、次にこの結果をそのオペランドの1つの欄に、或は出力バスの1つの更に別の欄に挿入する。FOU動作はADD、SUBTRACT、INCREMENT、DECREMENT、NEGATE(演算)、AND、OR、XOR、NOT、MAX、及びMINを含む。FOUは複数の独立した欄が同一語にパックされたパックド構造(packed structures)の高速操作を行うことによってシミュレーションアプリケーションを加速する。1つのサイクルにおいて、FOUは2つの語から欄を抽出し、これら欄に関して操作し、その結果を1つの語に挿入する。可能な欄の幅は1、2、4及び8ビ

ットである。バスA、B及びCのグルーピングから、FOUがバスA及びBにのみ接続されることがわかる。但し、FOUはレジスタアレイ73からのバス63にも接続され、この場合、レジスタR4及びR5がアクセスされる。

FOUはこの入力の1つをバスからではなく、アレイ73内の専用レジスタから取ることもできる。FOUはバス64を介してレジスタR15にアクセスできる。これは同一サイクルにおいても1つ多くのバス関連操作を行なうことを可能にする。

第6図はFOUユニットのブロック図を表わす。このユニットへの1つの入力は復号器84から来る。これは遂行されるべき所望の動作を指定するFOU-OP 4ビットバス；第1のオペランドのソースを決定するFOU-ASELライン、つまり、AバスあるいはレジスタR15に接続されたバス(バス64)；着信先バスを指定するFOU-OSELライン；及び信号をバス63に配るためにレジスタR4或は

R5のいずれかを用いるかを指定するFSR-SELラインを含む。バス83は回路を通じて用いられるA-START、B-START、OUT-START、IN-SIZE、及びOUT-SIZE制御信号を供給する。FOUユニットへの他の入力Aバス、Bバス、及びバス84から来る。

Aバス及びバス84はセレクト201に加えられ、ここで、1つがFOU-ASEL信号の制御下において選択される。セレクト201の出力及びBバスはラッチ回路202に加えられ、回路202はベーシッククロック(basic clock)の第1のフェーズによって制御される。このラッチされたBバスは入力セレクト203に加えられ、一方、もう一方のラッチされたバスは入力セレクト204に加えられ、セレクト203及び204は所望の欄をIN-SIZE制御信号及びA-START及びB-START制御信号の制御下で抽出する。後者は抽出された欄の開始ビットを決定し、一方、前

者は欄の長さ(1、2、4又は8ビット)を決定する。

セレクト203及び204の出力は機能ブロック205に加えられ、これは上に同定されたFOU動作の全て(例えば、ADD及びINCREMENT)を実現する。これら動作はオブコード及び駆動論理ブロック(opcode and drive logic block)206から派生される機能制御信号の制御下で実行される。ブロック206は複号器84によって供給される信号に応答し、これは従来の組合せ回路から成る。機能ブロック205は、同様に、従来の組合せ回路から成る。機能ブロック205の出力は出力挿入論理207に加えられる。制御信号OUT-SIZE及びOUT-STARTに응答し、ブロック207はブロック205によって生成された欄を所望の位置に挿入し、これを出力セレクト208及び209に転送する。セレクトは複号器84からの制御信号FOU-AEN及びFOU-BENに응答し、この出力を夫々A

BUS(77)及びB BUS(78)に転送する。

異なる長さの欄に関して操作するFOUと正反対に、AAUユニット(75)は16-ビット演算を遂行する。フェーズ1において、2つのソースオペランド、A及びCが指定のレジスタ、つまりキューユニット76及び/或はRAMインタフェース72からAバス(77)及びCバス(79)上に読み出される。これらはAAUへの入力である。AAU動作の結果はフェーズ3においてCバスにリターンされる。AAU命令は、ADD、SUBTRACT、INCREMENT、DECREMENT、MULTIPLY、DIVIDE、NEGATE(演算)、AND、OR、XOR、NOT、BIT TEST、BIT SET、BIT CLEAR、及びROTATEを含む。

AAUもまたその入力の1つをバスからでなくアレイ73内の専用レジスタから取る。AAUはレジスタR13にバス85を介してアクセ

スすることができる。これは同一サイクルにおいても1つ多くのバス関連動作を行なうことを可能にする。

フェーズ3におけるレジスタの書き込みはフェーズ2において遂行されたAAU動作の結果に依存させることができる。これはPE内での唯一の条件付実行メカニズムである。条件付分岐は、上に説明のごとく、ターゲットアドレスを2つのレジスタに書き込むことによって遂行される。

第7図はアドレス演算ユニット75のブロック図である。ユニットへの1つの入力複号器85から来る。そしてこれは遂行されるべき動作を指定する信号AAU-OP;シフティングを指定する信号AAU-SH;及びAバス或はレジスタR13(バス85)に接続されたバスのいずれかが選択されるべきかを指定する信号AAU-ASELを含む。

バス85及びAバスはAAU-ASEL信号によって制御されるセレクト211に接続され

る。選択されたバスは、Cバスと同様にラッチ212に加えられる。ラッチされたCバスはセクタ231及び214に接続され、ラッチされたAバス（或はバスB5）はセクタ214に接続される。セクタ213の出力はシフト論理ブロック215に接続される。ブロック213、214及び215はAAUのBIT TEST、BIT SET、及びBIT CLEAR動作、並びにAAUのSHIFT及びROTATE動作を実現する働きをする。このビット操作動作はA BUS（これはOPCODE及び駆動論理ブロック216に加えられる）上の制御信号及び4つの最下位ビットに回答して生成されるマスクを採用することによって実現される。論理ブロック218はまた上に説明の復号器85の出力信号にも回答し、機能ブロック217に加えられる制御信号を生成する。機能ブロック217はセクタ214及びシフト論理215から入力信号を受信する。これは上に説明の残りのAAU動作を遂行する。

出すのに用いられる。

64-ビットPEマイクロ命令は2つのフォーマット、つまり、コンスタント(constant)或はノーマル(normal)をもつ。コンスタントマイクロ命令はバスA及びB上に2つの16-ビットコンスタントを置き、コンスタントをレジスタにロードするのに用いることができる。但し、コンスタント命令の際にAAU或はFOU動作が遂行されることはない。ノーマルマイクロ命令は全ての値のケースにおいて用いることができ、以下を指定する欄を含む。

1. 3つのバスの各々のためのソースレジスタ及び優先レジスタ。短い(5-ビット)コンスタントもまたバスA上にマイクロ命令から直接に置くことができる。
2. AAU及びFOU動作。
3. アドレスレジスタの選択。
4. トラップ起動。
5. 条件選択及び個々のバスのこの条件へのセンシタイゼーション(sensitization)。

この機能は従来の組合せ論理によって実現される。最後に、ブロック21の出力は出力セクタ218に加えられ、セクタ218は復号器B5からのAAU-BN信号に回答する。ブロック217は又情報をブロック219に供給するが、ブロック219は専用旗置状態(specific flag conditions)の存在を確かめる。これら状態はAAU-BN信号に影響を与える。これら状態には、動作結果の信号を示すAAU-SIGN; AAUキャリー状態であるAAU-CARRY; オーバーフロー状態であるAAU-OVF; バスA及びCが同一情報を選ぶことを示すAAU-ABQC; 及び動作結果がゼロであることを示すAAU-ZEROが含まれる。

マイクロプログラムRAM(71)はデータ経路の個々の部分を制御するマイクロ命令を提供する。個々のサイクルにおいて、選択されたプログラムアドレスレジスタが次のサイクルにおいて遂行されるべきマイクロ命令を読み

MARSを多重遅延論理シミュレータとして用いることによって、個々のクラスタによって最大64Kゲートまで扱うことができる。64Kゲート以上が存在するような用途においては、第1図に従って、1つ以上のクラスタを採用することができる。ここで、個々のクラスタは回路のあらかじめ選択された部分のシミュレーションに昇げられる。勿論、回路の区画はクラスタ間の通信が最小化されるように選択される。

クラスタ内のシミュレーションはシーケンスのタイムステップ及び2つのシミュレーションフェーズ:つまり、ファンアウト(fanout)及び評価(evaluation)フェーズにて遂行される。ファンアウトシミュレーションフェーズの際に、現時間ステップにおいて起こる事象がゲートの入力に伝搬され、これらゲートが評価のためにスケジュールされる。次に評価フェーズにおいて、これらゲートがシミュレートされる。2つのシミュレーションフェーズの使

れはまたこの入力番号を出力ログ ブロック 104に送る。

ブロック104は監視されている信号の事象を記録する。これら信号は所望のゲート出力を特定するビット ベクトル(1ビット/ゲート)によって特定される。

ポインタ リスト ブロック105は出力ログ104からゲート値メッセージを受信し、そのゲートのファンアウト リスト内のポインタを調べ、ポインタ及びこの値をファンアウト リスト108に送る。このポインタは18ビットである。

ファンアウト リスト ブロック106はポインタ リストからポインタ105及び値メッセージを取り、このポインタをそのゲートに対するファンアウト リストを調べるために用いる。個々のファンアウト項目はゲートと入力番号のペア(gate and input number pair)から成る。個々のファンアウトに対して、入力メッセージからのこの値がこの項目に附加され、これ

メッセージが伝搬すると、評価フェーズが開始され、結果として、ゲート スケジューラ(第8図の右側の上部ブロック)はそのスケジューラされたゲートを入力テーブルに送る。

評価フェーズにおいて(第8図の右側部分)ゲート スケジューラはそのスタックからスケジューラされたゲートを取り出し、ゲートをスケジューラされていないとマークし、ゲート メッセージを入力テーブル ブロック107上に送る。最後のゲートがスタックから取り出されると、ゼロ メッセージが入力テーブルに送られる。

この入力テーブルはゲート メッセージを受信し、ゲートに対する現入力値を調べる。受信された個々のゲート メッセージに対し、ゲート識別子及びその入力値を含む2語メッセージがゲート タイプ テーブル112上に送られる。ゼロ メッセージは入力テーブルをファンアウト フェーズに戻す。

ゲート タイプ テーブルはゲート タイプ

によって入力テーブル107に送られる1つのゲート入力及び値メッセージ(gate input and value message)が形成される。

入力テーブル ブロック107は個々のゲートに対する入力値のリストを保持する。これがファンアウト リストから1つのゲート値及び入力メッセージを受信すると、テーブルをこれに従って更新し、そしてゲート識別子を含むメッセージをゲート スケジューラ108に送る。

ゲート スケジューラ ブロック108はその入力が評価(evaluation)に対して変更されたゲートをスケジューラする。入力テーブルから受信された個々のメッセージに対して、このゲートはこれが既にスケジューラされているかチェックし、されていない場合は、スケジューリング スタック上に押される。スケジューラされたゲートを追跡するためにビット ベクトルが用いられる。

ブロック101-108を通じてゼロ メ

ッセージを調べ、この情報を入力テーブルから受信されたメッセージに附加する。結果としての2語メッセージが機能ユニット111に送られる。

機能ユニット ブロック111は指定されたゲート機能ユニットをシミュレートし、入力メッセージ内のゲート タイプ及び入力値を用いてそのゲートによって生成される出力値を計算する。結果としての出力値は、ゲート識別子とともに、遅延テーブル110に送られる。

遅延テーブル110はゲート及び信号の変化の方向によって正しい遅延を調べる。こうして決定された遅延がゲート値メッセージに附加され、そして入力ベクトル リスト ブロック109にパスされる。

入力ベクトル リスト ブロック109は遅延を時間に変換し、メッセージを出力フィルタ(102)にパスする。これは次に現時間がそれに対して入力ベクトルがスケジューラされた最後の時間に達したか否かチェックする。現時間がこのベクトル時間以上である場合は、次の

スケジューラされたベクトルがメッセージ流内
に挿入され、出力フィルタ上にパスされる。

この出力フィルタはゲート評価の結果がゲート
出力の遅延を変えたか否か調べる。変えた場
合は、出力フィルタが未決事象をログし、ゲート
及び時間メッセージを信号スケジューラに送
くる。

信号スケジューラは出力フィルタから受信さ
れる事象をスケジューラする。ゼロ メッセ
ージによって、サイクル カウントがゼロでない
ことを条件として、次のファンアウト サイクル
が開始される。

4. 図面の簡単な説明

第1図は8個のクラスタが個々のノードに接
続され、別個のバスを通じて1つのホストコン
ピュータと通信する3-キューブ通信網を示し
;

第2図は個々のノードの一般構造を示し;

第3図はクラスタを構成する処理要素及び処
理要素を相互接続するローカル メッセージ

スイッチを示すブロック図であり;

第4図は第3図に示されるローカル メッセ
ージ スwitchの詳細なブロック図であり;

第5図は処理要素自体の詳細なブロック図で
あり;

第6図はF O Uユニットのブロック図であ
り;

第7図はA A Uユニットのブロック図であ
り;そして

第8図は論理シミュレーションを実現するた
めのクラスタの使用を図解する。

＜ 主要部分の符号の説明 ＞

- 1 0 ----- クラスタ
- 1 1 ----- ホストコンピュータ
- 1 2 - 2 4 ----- 双方向ライン
- 2 5 ----- バス
- 3 0 ----- クロスバースイッチ
- 3 2 ----- 出力バッファ

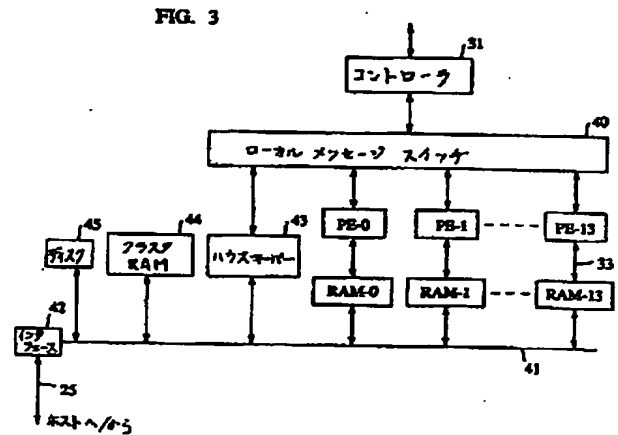
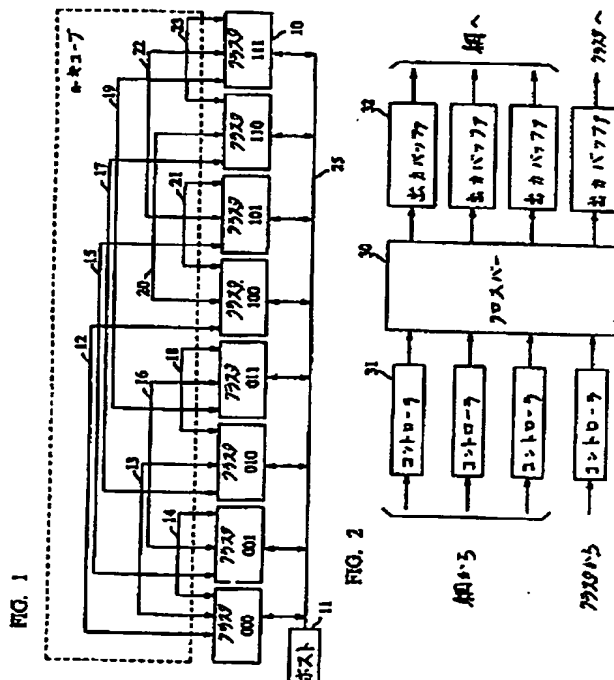


FIG. 4

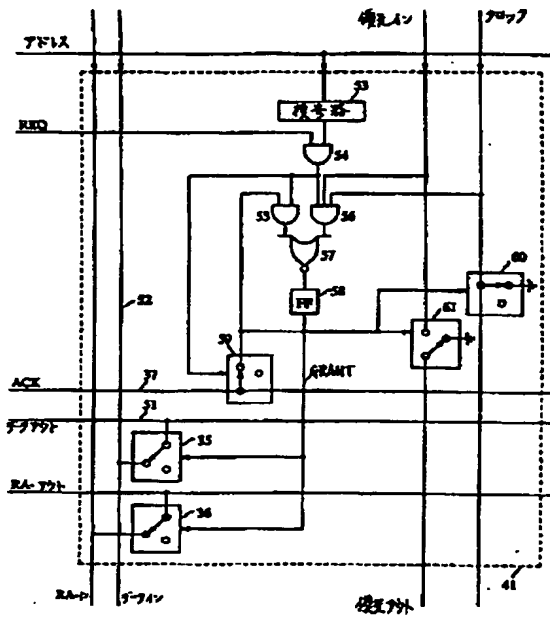


FIG. 5

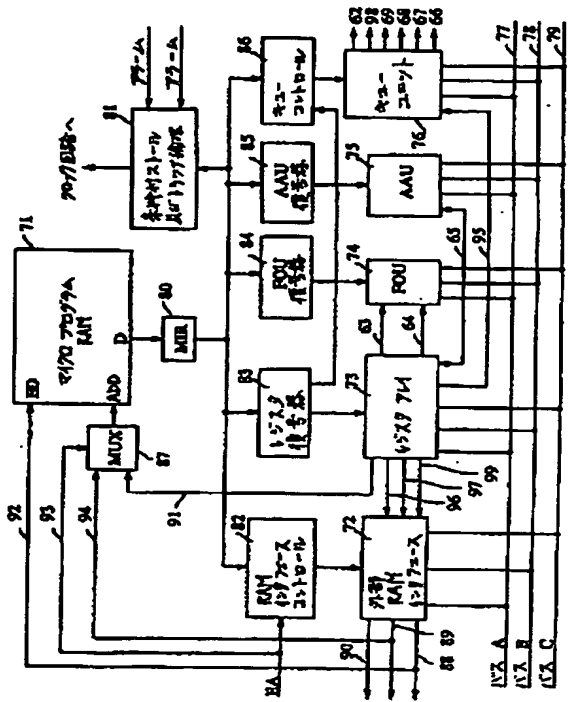


FIG. 6

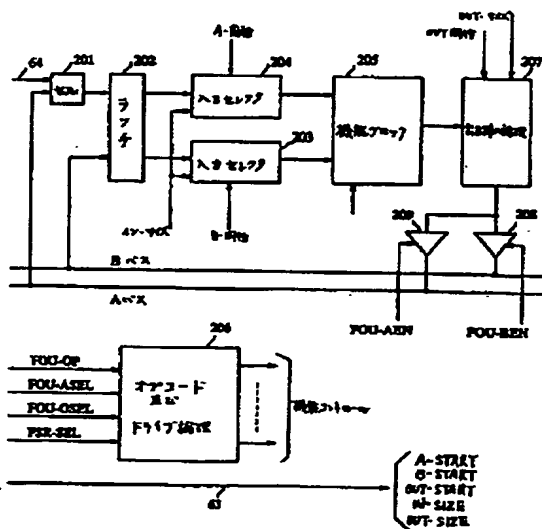


FIG. 8

